Summer Research Project: Quantifying Reliability using Adversarial Regions

Author: Ryan La
Project Partner: Yuye Zhang
Supervisors: Joerg Wicker & Katharina Dost
Public Code Repository: *github.com/ryanla-bme/Summer-Research-2021-Adversarial-Regions*

1 Introduction

Machine learning models are built on the assumption that the training data closely matches the data the model will be applied to. However, in reality, this assumption is often violated, resulting in a poor real-world performance for these models [1]–[4]. A cause of this violation is selection bias. This is when there is a distributional difference between training and real-world application data. Selection bias is caused by factors such as the inability to sample from regions in dataspace or over-representation. For example, due to equipment limitations [1] and remoteness of areas on an island [2].

The existing state of the art method to correct bias is Domain Adaptation (DA) which has shown success in fields such as computer vision and natural language processing [5]–[7]. DA takes advantage of an unlabelled dataset that is assumed to be unbiased; using frameworks such as Instance Weighting [6] or Adversarial Domain Alignment [5], DA shifts the biased training datasets distribution to resemble the unbiased unlabelled dataset.

What if we do not have this unbiased unlabelled dataset, such as in medicine, where data is scarce? Can we still find a way to correct the bias without this unlabelled dataset and no knowledge of where this bias is? This report presents our attempt to solve this problem, the Active Learning using Poisoning Attacks with Density Regularisation (ALPADR) framework. This framework leverages adversarial regions found through a poisoning attack loss function and regulation from a density function. The main intuition behind this framework is that selection biases cause poor performance by shifting the decision boundary. Poisoning attacks find regions in dataspace that can influence the decision boundary; hence adding data points in these dataspace regions to the biased dataset could potentially correct the influence of selection bias. The density regularisation prevents adding outliers (sparse regions) and to regions where there is already information (dense). Although this relies on the assumption that we can add data points, the hope is that it requires fewer points than DA and does not require knowledge of where the bias is. Also presented is a possible framework leveraging adversarial regions using Generative Adversarial Networks (GANs).

The report is structured as follows: Section 2 details the definition of selection bias used for this report and the problem statement, Section 3 explains how synthetic datasets with biases were generated, the ALPADR framework and experiments to investigate the viability of this framework, Section 4 presents the results of the experiments, Section 5 discusses these results in the context of the problem statement, and Section 6 presents a possible future direction with the GAN framework for bias correction.

2 Problem Statement

This report uses a definition of selection bias that is simple. For this report, selection bias is defined as $P(y_b|x_b) \neq P(y_T|x_T)$, where y_b, y_T, x_b and x_T are the labels and features of a biased training set D_U and unbiased test set D_T (real-world data the model will be applied to) respectively. The conditional probability densities are different for D_U and D_T. For examples of more formal detailed definitions of selection bias see works such as [8] and [4].

With this definition of selection bias, the problem statement we attempt to solve with the methods detailed in this report is as follows:

From a two-class labelled dataset $D_U = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, where (x_i, y_i) denotes a feature label-pair, a biased labelled subset ($D_B \subseteq D_U$) is drawn. Let D_T be an unbiased testing set drawn from the same distribution as D_U. Given only D_B and without any knowledge about how the biased distribution of D_B differs

from that of the unbiased dataset D_U, can we correct the selection bias. In other words, ***our goals are can we: (a) minimise the difference between the performance on an unbiased testing set D_T on a classifier trained on D_U and D_B, (b) identify how D_B is biased (or differs from D_U).*** Note although we only use Gaussian distributions for the datasets in this report, this problem statement generalises to any distribution for D_U, normal or otherwise.

3 Methods

To explore the viability of our proposed framework as a solution to the problem statement, we experiment with synthetic toy datasets with an induced selection bias. This section details how these synthetic datasets were generated and how selection bias was induced in these datasets. The proposed framework of Active Learning using Density Regulated Poisoning Attacks (ALPADR) is then explained with the intuition behind it.

3.1 Synthetic Dataset Generation

The synthetic datasets used in the experiments were 2-class and 2-dimensional. Both classes were generated as multivariate Gaussian distributions with different parameters. This can be described by the following equation,

$$X_\theta \sim N(\mu_\theta, \Gamma_\theta), \tag{1}$$

where $\theta \in \{0,1\}$ represents the class, X_θ are datapoints for class θ, μ_θ are the means for the datapoints in class θ and Γ_θ is the covariances for class θ. Figure 1 shows an example of a synthetic dataset generated by the method described which is the same as the D_U for one of our experiments. The parameters for this dataset were $\mu_0 = \begin{bmatrix}-1.5\\0\end{bmatrix}$, $\Gamma_0 = \begin{bmatrix}1 & -0.5\\-0.5 & 1.5\end{bmatrix}$, $\mu_1 = \begin{bmatrix}1.5\\0\end{bmatrix}$ and $\Gamma_1 = \begin{bmatrix}1 & 0.5\\0.5 & 1.5\end{bmatrix}$. 300 samples were generated for each class. This dataset was trained on a linear SVM and tested against another dataset sampled with the same parameters but with 1000 samples drawn from each class. Synthetic Gaussian datasets were chosen as they are very common in real world datasets, as well as many datasets being able to be approximated by Gaussians [9].

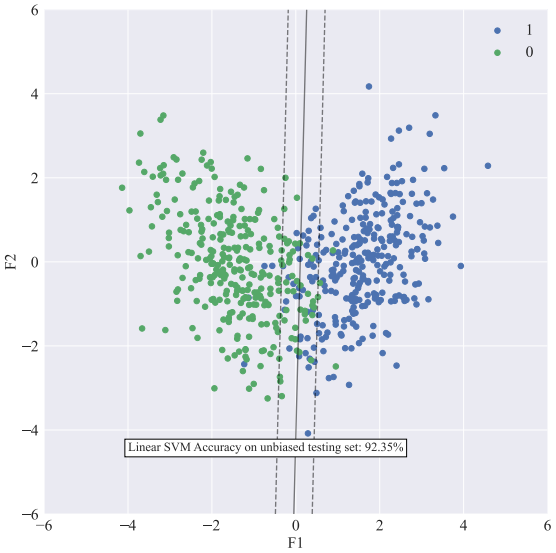

Figure 1. Example of synthetic dataset used in experiments. The axes labels of F1 and F2 denotes two arbitrary features, the green and blue points represent the datapoints for class 0 and 1 respectively. The plot also includes the decision boundary of a linear SVM trained on this dataset.

3.2 Bias Generation

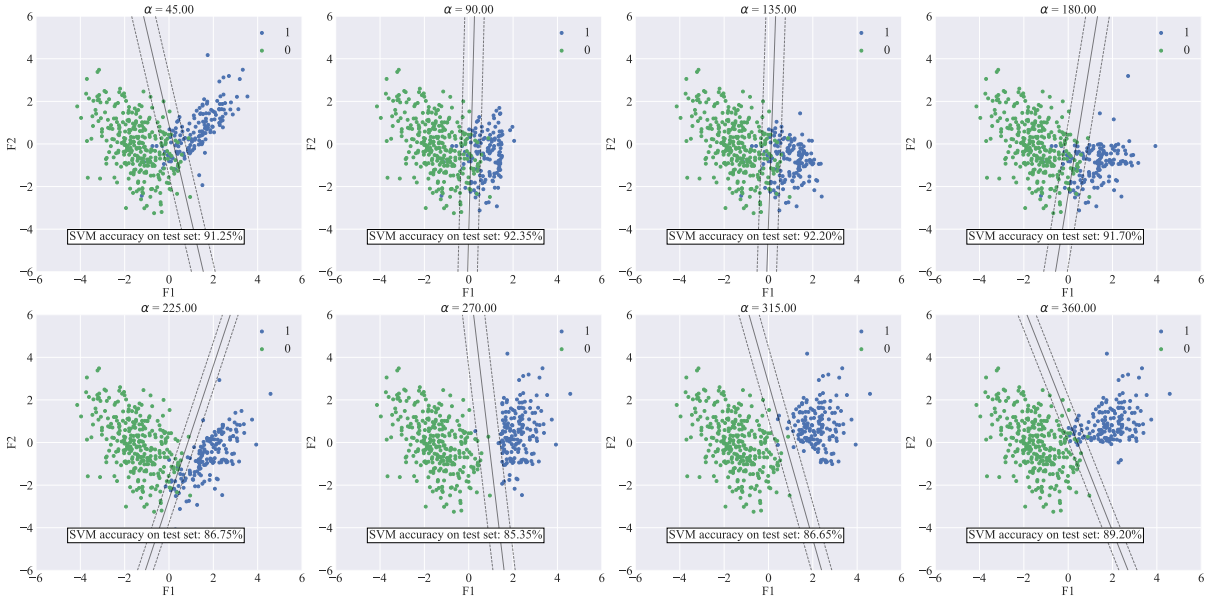

Figure 2. Examples of different biases induced in the synthetic dataset from Figure. 1. A linear SVM was trained on these datasets and tested against the same testing set as in Figure 1. For all these biases $C = \begin{bmatrix} 0 \\ 1.5 \end{bmatrix}, p = 0.05$ *and the biases were induced in the datapoints for class 1.*

Selection bias was induced in one of the classes of a synthetic dataset by selecting an angle of rotation α (degrees) and a translation matrix C. The datapoints x in the selected class are rotated clockwise by α and translated by -C. Let x' denote these rotated and translated points. The x which have it's x' with the second feature $F_2 \geq 0$ is kept and if $x'\ F_2 < 0$ then x is kept in the dataset with probability p. In other words, datapoints x 'below' a plane with a rotation angle of α and centre of C are kept in the dataset with probability p. Figure 2 shows multiple selection biases induced in the dataset generated in Figure 1.

Intuition: Inducing a selection bias in this method relates to real world examples where access to a certain region in dataspace is restricted. For example, stars in distant galaxies [1] and areas with difficult terrain [2]. Resulting in little or no data in that region of dataspace.

3.3 Proposed Framework – Active Learning using Poisoning Attacks with Density Regularisation (ALPADR)

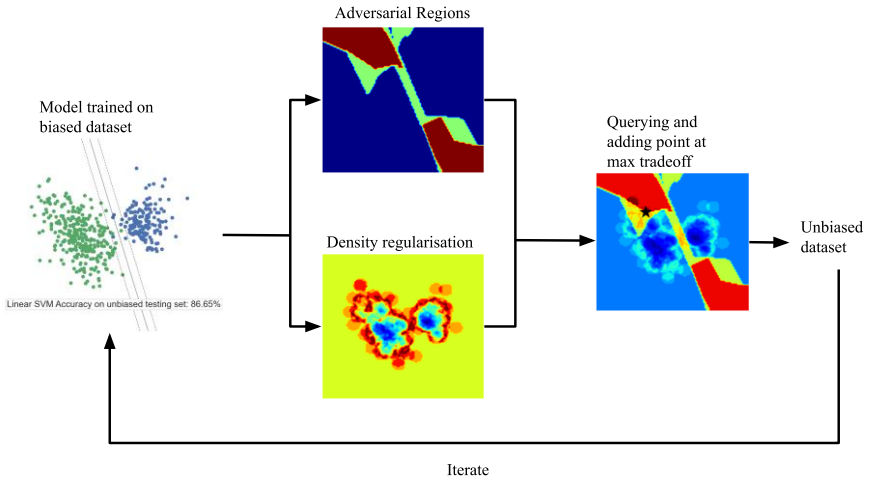

Figure 3. Block diagram overview of ALPADR.

The fundamental idea of how ALPADR corrects bias is a framework that iteratively adds datapoints in regions of dataspace where the selection bias is or can help correct the selection bias. **Querying** is when a datapoint is labelled by an Oracle and added to the dataset. The **Oracle** for active learning is usually an expert in the field of the dataset. This process of iteratively selecting a point in dataspace according to some criterion, querying and adding to the dataset is known as Active Learning [10]. As we are using a synthetic dataset the Oracle used was a linear SVM trained on D_T. To choose the datapoint to query add to the biased dataset, ALPADR uses a trade-off between a poisoning attack loss and a density regularisation function. This is described by the following,

$$\underset{x_p}{argmax}\, H\left(x_p, D_B\right) = L\left(x_p, D_B\right) + \lambda \cdot N\left(x_p, D_B\right), \tag{2}$$

where x_p is a point in data space, D_B is a biased dataset, L is a loss function, N is a density function and λ is a trade-off parameter. The mathematical form for Equation (2) resembles a Tikhonov Regularisation scheme. So that it is easier and more explicit to set the importance between L and N, both were normalised (e.g., if $\lambda = 1$, L and N have the same importance in the trade-off, if $\lambda = 0.5$ then N is 50% as important in the trade-off).

$$L_{norm}\left(x_p, D_B\right) = \frac{L\left(x_p, D_B\right)}{L_{max}\left(X_p, D_B\right)}, \tag{3}$$

$$N_{norm}\left(x_p, D_B\right) = \frac{N\left(x_p, D_B\right)}{N_{max}\left(X_p, D_B\right)}. \tag{4}$$

$L_{max}\left(X_p, D_B\right)$ is the maximum loss over all points in a bounded dataspace with $N_{max}\left(X_p, D_B\right)$ similarly. Equation (2) now becomes,

$$\underset{x_p}{argmax}\, H\left(x_p, D_B\right) = L_{norm}\left(x_p, D_B\right) + \lambda \cdot N_{norm}\left(x_p, D_B\right). \tag{5}$$

For the rest of the report L and N will refer to the normalised versions L_{norm} and N_{norm} unless stated otherwise. Equations (2) and (5) can generalise to any appropriate loss function for L and density function for N. The x_p that optimises Equation (2) was found using Grid Search on a bounded region in dataspace. The exact L and N functions used for the implementation of ALPADR in this report are detailed below.

Poisoning attack loss function: The L function used for ALPADR is a simple accuracy loss [11] described by the following,

$$L\left(x_p, D_B, F\right) = 1 - Acc\left(x_p, D_B, F\right). \tag{6}$$

$Acc\left(x_p, D_B, F\right)$ is calculated as follows where F is a classifier,

1. D_B is 50-50 split into a training and validation set. Let these be $D_{B,train}$ and $D_{B,val}$ respectively.
2. $Acc\left(x_p, D_B, F\right)$ is the the minimum accuracy of F on $D_{B,val}$ out of F trained on $D_{B,train} \cup x_p$ with x_p label as 0 or 1.

Density regularisation function: The N function used for ALPADR is log negative quadratic function described by the following,

$$N\left(x_p, D_B, \sigma, \nu\right) = -1 \cdot ln\left|\left(A\left(x_p, D_B, \sigma\right) - \nu\right)^2 + 1\right| \tag{7}$$

where A is a function that calculates the number of data points around x_p within a Euclidean distance of radius σ. ν is the parameter that shifts the maximum of N.

ALPADR intuition: Poisoning attacks find points in dataspace that where if datapoints (mislabelled or otherwise) were added into the dataset, would influence the decision boundary to change. The potential connection with selection bias is that selection bias also causes a change in decision boundary (Figure 2), but in the 'opposite' manner, by 'removing' datapoints. So, if we add datapoints using a poisoning attack this could potentially reverse the effect of the selection bias.

The intuition behind using density regularisation is that we do not want to add outliers or points in dense areas of dataspace where there is already information. Hence, we want to down weight regions of dataspace where that is very sparse as well as very dense. Therefore, we chose Equation (7) for N.

3.4 Experiments

The two goals stated in the problem statement were:

a) **Minimise the difference between the performance on D_T for a classifier trained on D_U or D_B.**
b) **Identify how D_B is biased (or differs from D_U).**

Experiments were performed to investigate the potential of ALPADR to solve these goals. Goal **(a)** was investigated by performing 100 iterations of ALPADR and observing the change in accuracy on D_T. The experiments were also compared with baselines of Uncertainty sampling and Random Ideal sampling. Goal **(b)** was investigated by observing where ALPADR was querying and adding points.

Two different two-class multivariate Gaussian datasets were used for the experiments which are shown in Figure 4 and their parameters summarised in Table 1. Biases were induced for class 1 in these datasets with $\alpha = [225,270,315,360]$ as these appeared to have the most effect on accuracy (Figure 2). Note when a bias was induced, datapoints were removed resulting in the number of datapoints in the biased class approximately half what is stated in Table 1 ($|D_U| > |D_B|$). The Oracle was a linear SVM trained on D_T, which has the same distribution as D_U except with 1000 datapoints per class instead of 300.

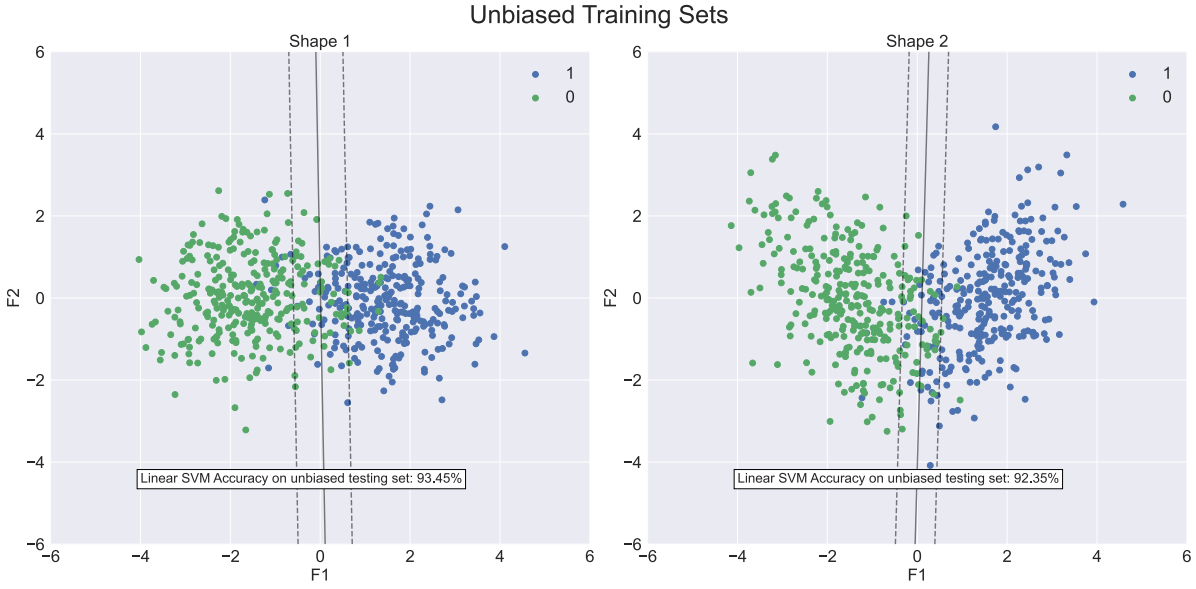

Figure 4. Unbiased synthetic datasets used for experiments trained on a linear SVM classifier and evaluated on D_T.

The parameters for Equation (7) was chosen heuristically ($\sigma = 0.5$ $\nu = 3$) to give an appropriate down weighting of dense and sparse regions in dataspace. ALPADR was applied to a dataspace that was bounded by a range of -6 to 6 for both features. Two trade-off parameters ($\lambda = 0.4, 0.8$) were used in the experiments to observe the effect of the importance of density. A linear SVM classifier trained on D_B was chosen as the machine learning model for these experiments as it is simple to interpret and identify if the bias has been corrected. ALPADR and the experiments were programmed in python and can be found in public code repository *github.com/ryanla-bme/Summer-Research-2021-Adversarial-Regions*.

Table 1. Parameters for datasets used in experiments.

	Means	Covariances	Number of datapoints generated for each class	Biases induced for class 1
Shape 1	$\mu_0 = \begin{bmatrix}-1.5\\0\end{bmatrix}$, $\mu_1 = \begin{bmatrix}1.5\\0\end{bmatrix}$	$\Gamma_0 = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$, $\Gamma_1 = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$	300	$\alpha = [225,270,315,360]$
Shape 2	$\mu_0 = \begin{bmatrix}-1.5\\0\end{bmatrix}$, $\mu_1 = \begin{bmatrix}1.5\\0\end{bmatrix}$	$\Gamma_0 = \begin{bmatrix}1 & -0.5\\-0.5 & 1.5\end{bmatrix}$, $\Gamma_1 = \begin{bmatrix}1 & 0.5\\0.5 & 1.5\end{bmatrix}$	300	$\alpha = [225,270,315,360]$

3.4.1 Baselines

The traditional goal of active learning is generally to find points in dataspace if added to the training set would improve the performance of the model the most [10]. Although this is not specifically targeting bias correction, they could still provide a useful baseline to assess ALPADR against. Uncertainty sampling and Random Ideal sampling were the two active learning strategies used as baselines.

Uncertainty sampling queries points in dataspace that the classifier is least confident in classifying. This is shown by,

$$\underset{x_p}{argmax}\, H\left(x_p, D_B\right) = 1 - max\, P(y_i, x_p), \tag{8}$$

where $P(y_i, x_p)$ is the probability, the classifier believes x_p is in class i where $i \in \{0,1\}$.

Random Ideal sampling is an optimistic random sampler where datapoints are queried by drawing samples from a multivariate Gaussian with the same distribution (means and covariances) as $\boldsymbol{D_T}$.

4 Results

The results shown in this report are only for the experiments with $\alpha = 270$ for the Shape 1 dataset and $\alpha = 315$ for the Shape 2 dataset. These were chosen as the illustrative results and including the rest of the results ($\alpha = [225,315,360]$ for Shape 1 and $\alpha = [225,270,360]$ for Shape 2) would not change our conclusions about the potential of ALPADR. The $\alpha = 270$ and $\alpha = 315$ biases tests if ALPADR can correct a bias that translates or rotates the decision boundary respectively (Figure 5). The rest of the results are included in the public code repository.

Figure 5. Shape 1 with bias $\alpha = 270$ translates the decision boundary and shape 2 with bias $\alpha = 315$ rotates the decision boundary with respect to unbiased decision boundaries in Figure 4.

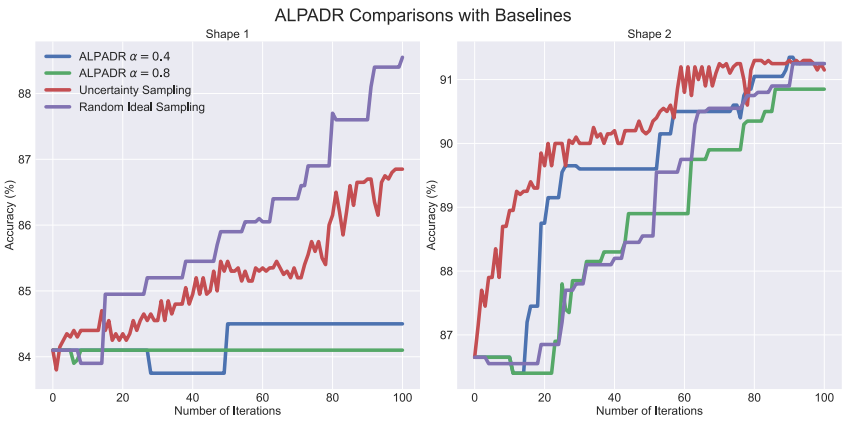

Figure 6. ALPADR comparison with baselines for 100 iterations. The bias induced was $\alpha = 270$ for the Shape 1 dataset and $\alpha = 270$ for the Shape 2 dataset.

From Figure 6 it does not appear that ALPADR is an improvement on existing active learning strategies. Indeed, for the Shape 1 experiment it performed worse than both baselines and there appeared to be minimal improvement in accuracy for ALPADR with $\lambda = 0.4$ and no improvement in accuracy for $\lambda = 0.8$. For the Shape 2 experiment ALPADR at least appeared to be on par with the baselines, converging towards the unbiased accuracy of 92.35% Figure 4.

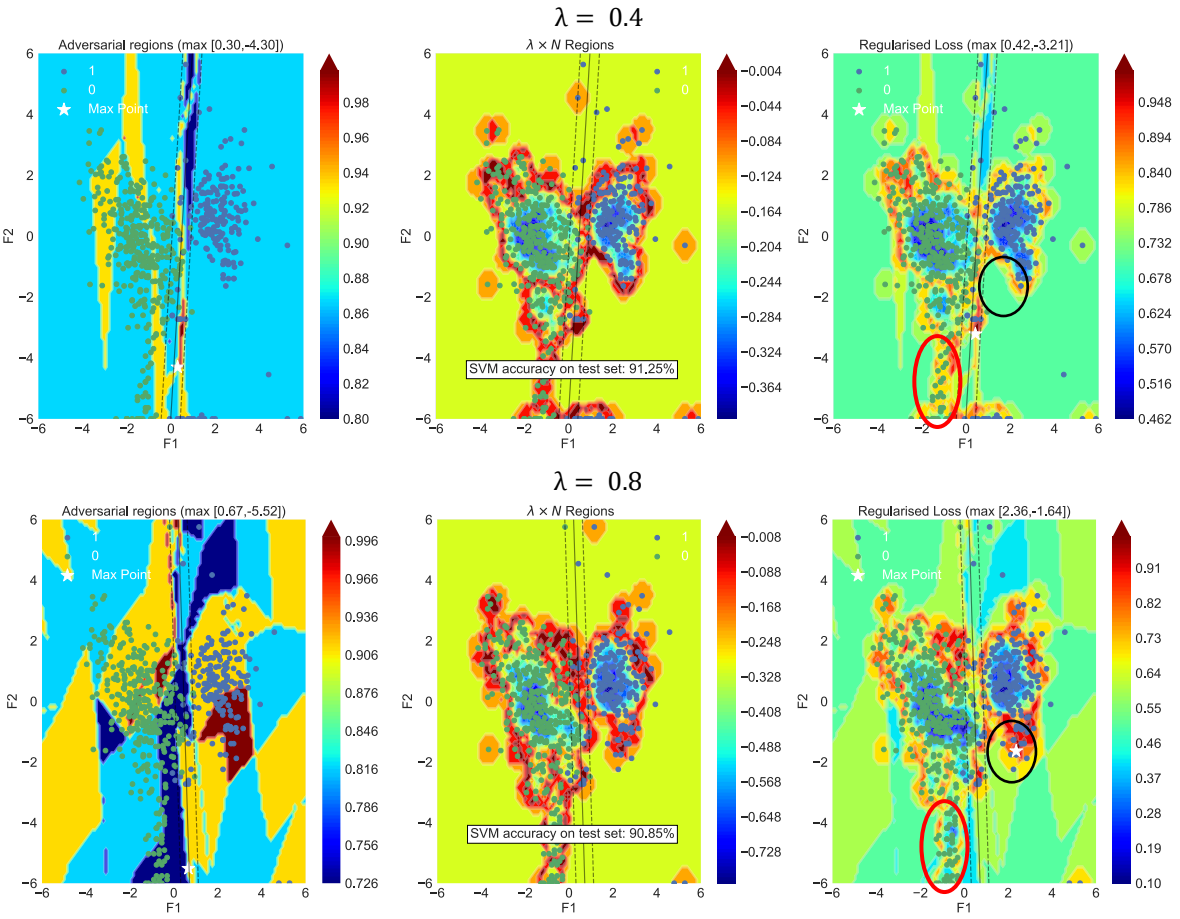

Figure 7. 100 iterations of ALPADR for the Shape 2 synthetic dataset for different trade-off (λ) parameters with $\alpha = 315$ bias induced. The left plots show the adversarial regions from the poisoning attack L function. The middle plots show the density N multiplied by λ. The right plots show the regularisation H function. The red circles on the right plots shows a region that has been queried unlikely to be from the original distribution. The black circles point to where D_B differs from D_U shown in Figure 4.

Observing Figure 7, it does appear that ALPADR adds points that are correcting the $\alpha = 315$ bias in the Shape 2 dataset so that the decision boundary is almost vertical and centred on $F1 = 0$ like as in Figure 4. Increasing λ from 0.4 to 0.8 appears to reduce the number of outliers queried and increases the number of points queried in the bias region shown by the black circle in Figure 7. However, the number of points queried in the bias region is still a small number. For both λ there are many points queried that are unlikely in the distribution shown by the red circles in Figure 7 for Shape 2 parameterised in Table 1.

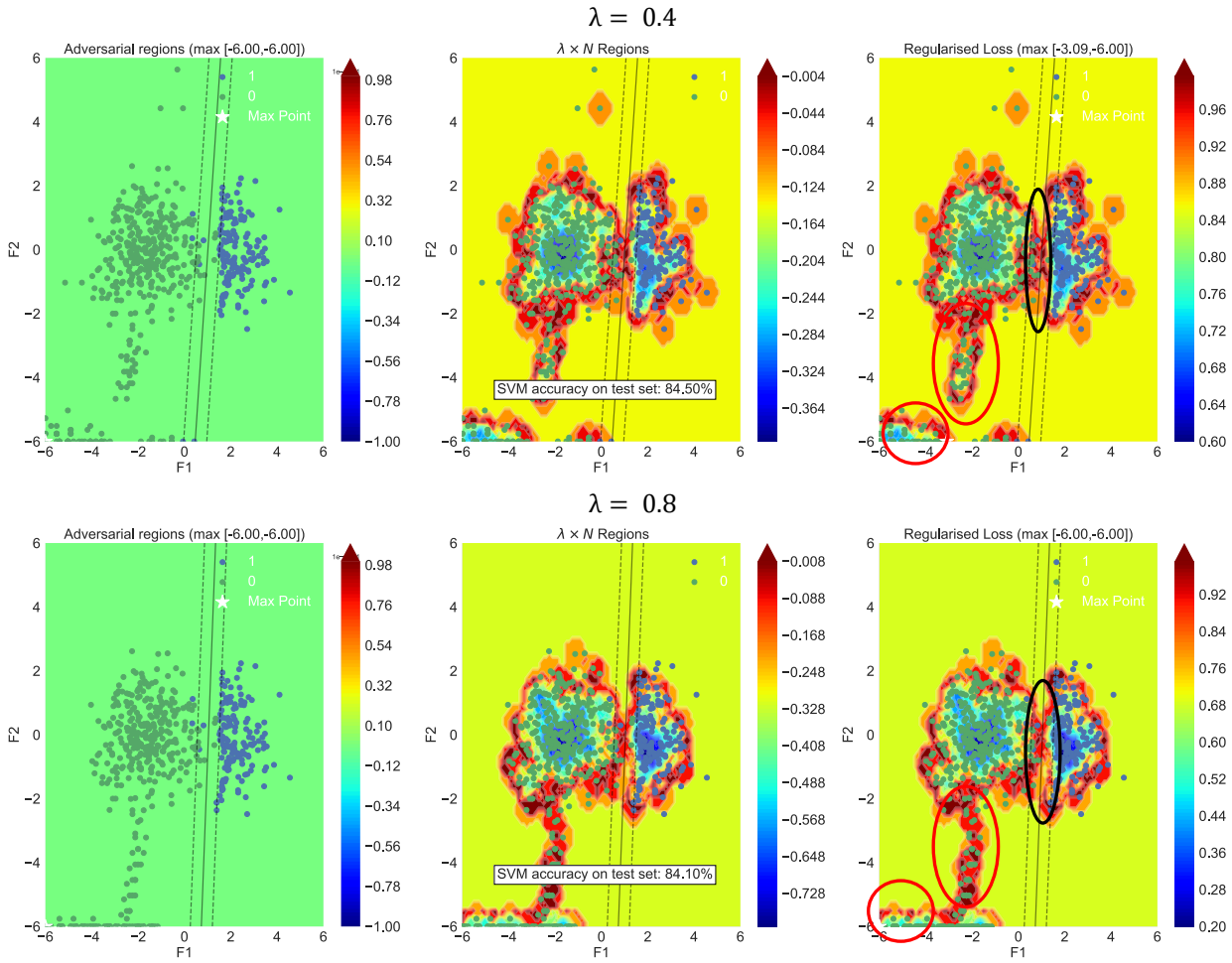

Figure 8. 100 iterations of ALPADR for the Shape 1 synthetic dataset for different trade-off (λ) parameters with $\alpha = 270$ bias induced. The left plots show the adversarial regions from the poisoning attack L function. The middle plots show the density N multiplied by λ. The right plots show the regularisation H function. The red circles on the right plots shows a region that has been queried unlikely to be from the original distribution. The black circles point to where D_B differs from D_U shown in Figure 4.

For the Shape 1 experiment shown in Figure 8 ALPADR does not appear to be querying points that corrects the bias. After 100 iterations there does not seem to be significant improvement in the movement of the decision boundary back to be centred around $F1 = 0$ like in the unbiased model shown in Figure 4. The queried points do rotate the decision boundary from Figure 5. Like for the Shape 2 experiment increasing λ from 0.4 to 0.8 reduces the number of outliers queried. However, there are still many points queried that are unlikely in the original distribution shown by the red circles in Figure 8. There also does not appear to be many points queried in the bias region shown by the black circle.

5 Discussion

From the results it does not appear that ALPADR reliably meets the two goals set out in the problem statement. For the Shape 2 experiment there appears to be improvement in performance on par with the baselines. The bias also appears to be corrected with the decision boundary rotated almost to vertical which meets goal (a). However, there are many points queried that are unlikely in the original distribution. This is a problem as if ALPADR were to be applied to a real-world application, these outlier points maybe difficult or even impossible for an Oracle to accurately label, making ALPADR infeasible for real world applications. There are also not many points queried in the bias region, therefore not meeting goal (b). This is also undesirable as this means the points queried by ALPADR may work to correct bias for a linear SVM, but these points are not transferrable and may not work to correct bias for other models. This also does not help with identifying where the bias region is, which may be important as it can help the user more systematically fix the bias. For example, if a bias

region is identified, the user could investigate and find it is due to faulty equipment. The user then subsequently fixes the equipment resulting in future data being unbiased.

ALPADR appears to struggle even more with the Shape 1 experiment, having minimal to no improvement in performance. This appears to be because ALPADR does not query points that translate the decision boundary, but rather prefer points that rotate the decision boundary. This observation is supported by the fact that the decision boundary is rotated due to the queried points but not translated for the Shape 1 experiment. This suggests that ALPADR in the state we presented in this report will not work on selection biases that translate the decision boundary. The same problems as in the Shape 2 experiment of querying points that are unlikely in the original and not many points queried in the biased region is also present in the Shape 1 experiment.

Perhaps the crux of why ALPADR does not reliably meet our goals for these synthetic datasets is because ALPADR's poisoning attack chooses the most influential single point in dataspace. However, as we can see in Figure 2, the selection bias causes a shift in decision boundary as an aggregate result of all the points removed and the selection biases also has a range of influences on the decision boundary. Hence it is not necessary and maybe even unlikely that the most influential points found with the poisoning attack coincide with the points in the biased region.

6 Future Work– cGAN for Bias Correction (CGBC)

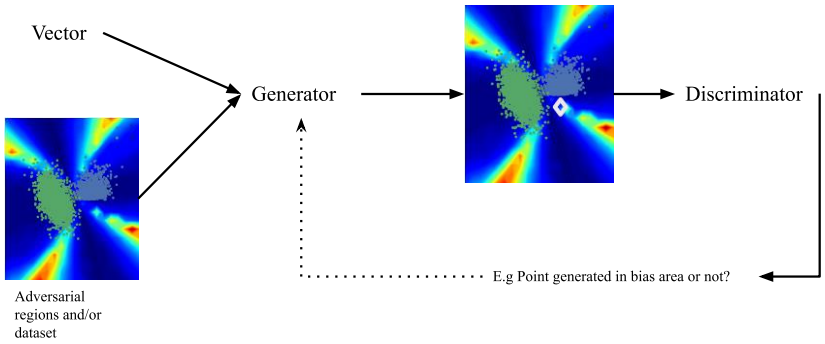

Figure 9. Block diagram overview of GCBC.

Figure 9 shows a block diagram of the CGBC framework to train a generator that can potentially correct bias. The GCBC framework follows the GAN framework popularised by [12]. The ultimate goal of the GCBC framework would be to train a generalised generator that given any biased dataset, either generate an unbiased dataset or identify where the bias region is. Concepts for the generator and discriminator for this framework are detailed below as well as GCBC's intuition.

Generator: The potential inputs to the generator trained by the GCBC framework are:

- The **adversarial regions**, which is a grid in dataspace of evaluations of a poisoning attack function such as Equation *(6)*.
- The **biased dataset**.
- Both the adversarial regions and biased dataset.
- **Random vector**. If the generator generates the unbiased dataset, then a random vector would allow different likely unbiased datasets to be generated.

Discriminator: The discriminator would then train the generator by using a loss function that determines how close the output of the generator is to an unbiased dataset or identifying the biased area.

cGAN intuition: Figure 10 shows the adversarial regions for different biases on the same dataset. Different biases appear to have different adversarial regions. These differences can potentially be exploited by the GCBC framework, and the unbiased version of these biased datasets be learned by the generator.

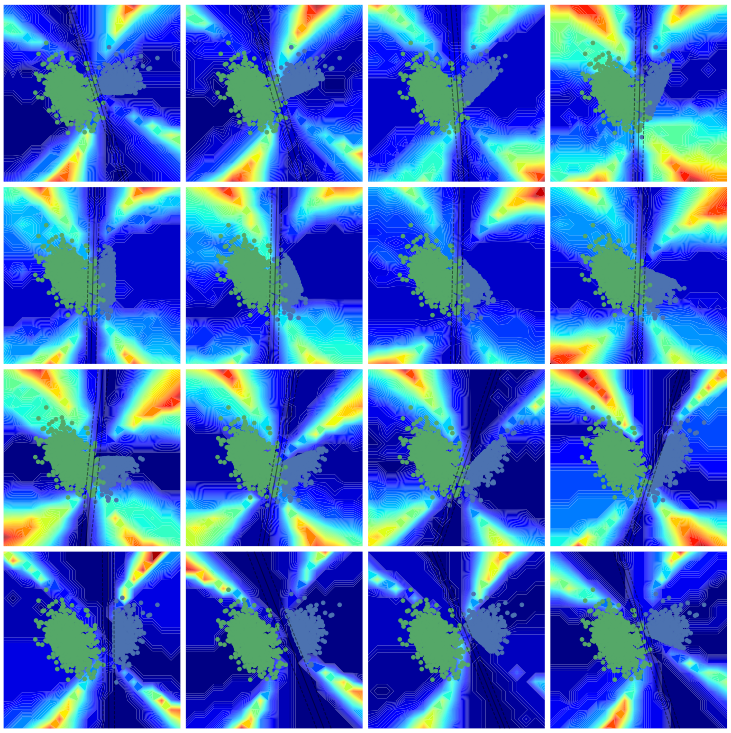

Figure 10. Adversarial regions for Shape 2 dataset with different biases. Generated similarly to Shape 2 experiments.

7 Conclusion

Experiments involving different biases and two synthetic datasets were conducted to investigate the viability of ALPADR in correcting bias. ALPADR did not appear to reliably correct bias, particularly struggling with a selection bias that translated the decision boundary. Numerous outliers were also queried and not many points in the bias area were queried in the experiments. Potentially the crux of why ALPADR does not reliably correct bias is due to points of highest influence found by ALPADR's poisoning attack function not coinciding with the points in the bias area, as the points in the bias area may not be the most influential. An alternative concept to correcting bias by training a generator with adversarial regions in a GAN framework was also presented (GCBC).

References

[1] J. W. Richards *et al.*, "ACTIVE LEARNING TO OVERCOME SAMPLE SELECTION BIAS: APPLICATION TO PHOTOMETRIC VARIABLE STAR CLASSIFICATION," *Astrophys. J.*, vol. 744, no. 2, p. 192, 2011, doi: 10.1088/0004-637x/744/2/192.

[2] S. Kramer-Schadt *et al.*, "The importance of correcting for sampling bias in MaxEnt species distribution models," *Divers. Distrib.*, vol. 19, no. 11, pp. 1366–1379, Nov. 2013, doi: https://doi.org/10.1111/ddi.12096.

[3] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A Survey on Bias and Fairness in Machine Learning," *ACM Comput. Surv.*, vol. 54, no. 6, Jul. 2021, doi: 10.1145/3457607.

[4] E. Bareinboim, J. Tian, and J. Pearl, "Recovering from Selection Bias in Causal and Statistical Inference," in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014, pp. 2410–2416.

[5] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial Discriminative Domain Adaptation," *CoRR*, vol. abs/1702.0, 2017, [Online]. Available: http://arxiv.org/abs/1702.05464.

[6] J. Jiang and C. Zhai, *Instance Weighting for Domain Adaptation in NLP*. 2007.

[7] C. Cortes and M. Mohri, "Domain adaptation and sample bias correction theory and algorithm for regression," *Theor. Comput. Sci.*, vol. 519, pp. 103–126, 2014, doi:

https://doi.org/10.1016/j.tcs.2013.09.027.

[8] K. Dost, K. Taskova, P. Riddle, and J. Wicker, “Your Best Guess When You Know Nothing: Identification and Mitigation of Selection Bias,” in *2020 IEEE International Conference on Data Mining (ICDM)*, 2020, pp. 996–1001, doi: 10.1109/ICDM50108.2020.00115.

[9] A. Lyon, “Why are Normal Distributions Normal?,” *Br. J. Philos. Sci.*, vol. 65, no. 3, pp. 621–649, Sep. 2014, doi: 10.1093/bjps/axs046.

[10] B. Settles, “Active Learning Literature Survey,” 2009.

[11] B. Biggio, B. Nelson, and P. Laskov, “Poisoning Attacks against Support Vector Machines,” in *Proceedings of the 29th International Coference on International Conference on Machine Learning*, 2012, pp. 1467–1474.

[12] I. Goodfellow *et al.*, “Generative Adversarial Nets,” in *Advances in Neural Information Processing Systems*, 2014, vol. 27, [Online]. Available: https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.